VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Data Visualization

460-4120

Fall 2024

Last update 9. 10. 2024

The Data as a Quantity

- Quantities can be classified in two categories:
  - Intrinsically continuous (scientific visualization, or scivis)
    - e.g. pressure, temperature, position, speed, density, force, color, light intensity etc.
  - Intrinsically discrete (information visualization, or infovis)
    - e.g. text, hypertext, content of web pages, database records etc.
- Sampled data – originally continuous data represented in a finite approximative form
- Corollary of the difference between sampled and discrete data:

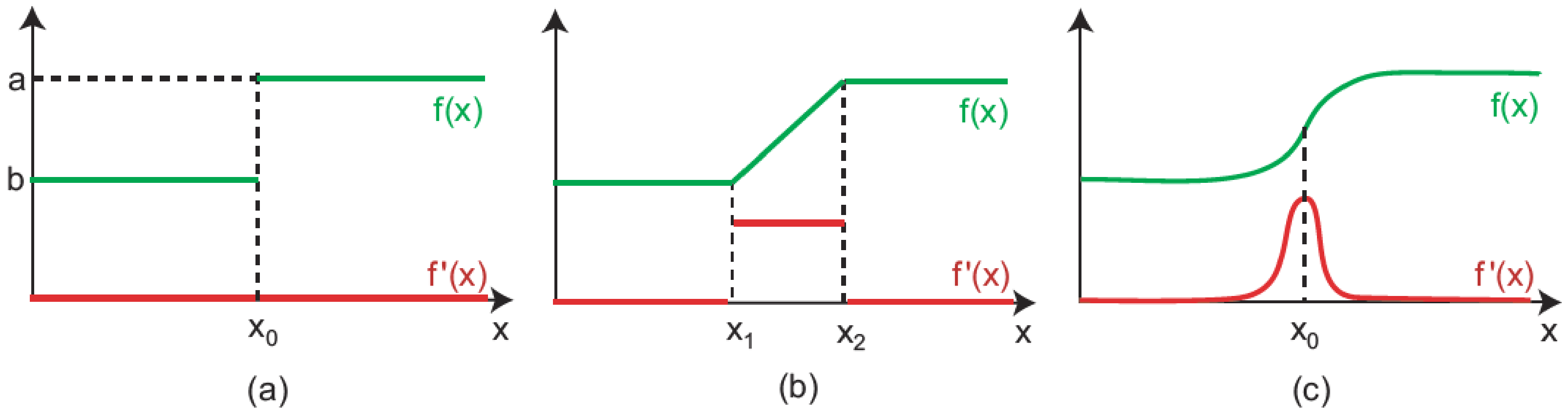
  In the case of sampled data, we can go back to a continuous approximation of the original (intrinsically) continuous data but it make no sense for (intrinsically) discrete data

Continuous Data

- Continuous data as a function:

$$f : D \rightarrow C$$

- Intuitive interpretation of continuity:



Discontinuous function | First-order C_0 cont. | High-order C_k cont.

Source: Data Visualization: Principles and Practice

Datasets and Dimensions

- Let the triplet $D = (D, C, f)$ define a continuous dataset
- We assume that $f: D \to C$
- D refers to a function domain, C is the function codomain
- d is the geometrical dimension of the space $\mathbb{R}^d$ into which D is embedded
- s represents the topological dimension of D itself (e.g. plane in the Euclidean space $\mathbb{R}^3$ has $s = 2$)
  - It holds that $s \leq d$
  - s is number of independent variables required to represent the domain D
- Codimension of an object of some d and s is the difference $d - s$

Datasets and Dimensions

- Virtually all data-visualization applications fix geometrical dimension to $d = 3$
- Only the topological dimension varies $s = \{1, 2, 3\}$
    - $s = 1$ corresponds to curves
    - $s = 2$ corresponds to surfaces
    - $s = 3$ corresponds to the volumetric datasets
- Topological dimension and dataset dimension are often used interchangeably
- Topological dimension is important in case of sampled datasets and when choosing a grid cell type

Datasets and Dimensions

- Function values are called dataset attributes
- The dimensionality c of the function codomain C is also called the attribute dimension
- Typically ranges from 1 to 4
- E.g. temperature assigned to some point in the Euclidean space $\mathbb{R}^3$ has $c = 1$ (it is a scalar value)

Sampled Data

- Typically, continuous functional representation of data is not available
- Moreover, several operations (filtering, simplification, analysis etc.) on continous data are not efficient
- Visualization applications work predominantly with sampled datasets
- Important operations relating continuous and sampled data:
    - Sampling – quite straightforward
    - Reconstruction – more complicated, the goal is to recover an approximated version of the original continuous data

Sampled Data

- Reconstruction employs interpolation of the values of the function between its sample points
- Two basic forms of sampling strategies:
  - Uniform
  - Non-uniform (e.g. respecting the distribution of the importance of the sampled data)
- Sampled dataset should be accurate (up to an user-specified error), minimal (w.r.t. an error), generic (operations), efficient (algorithmically), and simple (implementation) [Schroeder et al. 2006]

Sampled Data

- Sampled dataset $\{f_i, p_i\}$ consists of a set of N sample points and values
- Interpolation - the reconstructed function should equal the original one at all sample points, i.e. $\tilde{f}(p_i) = f(p_i) = f_i$

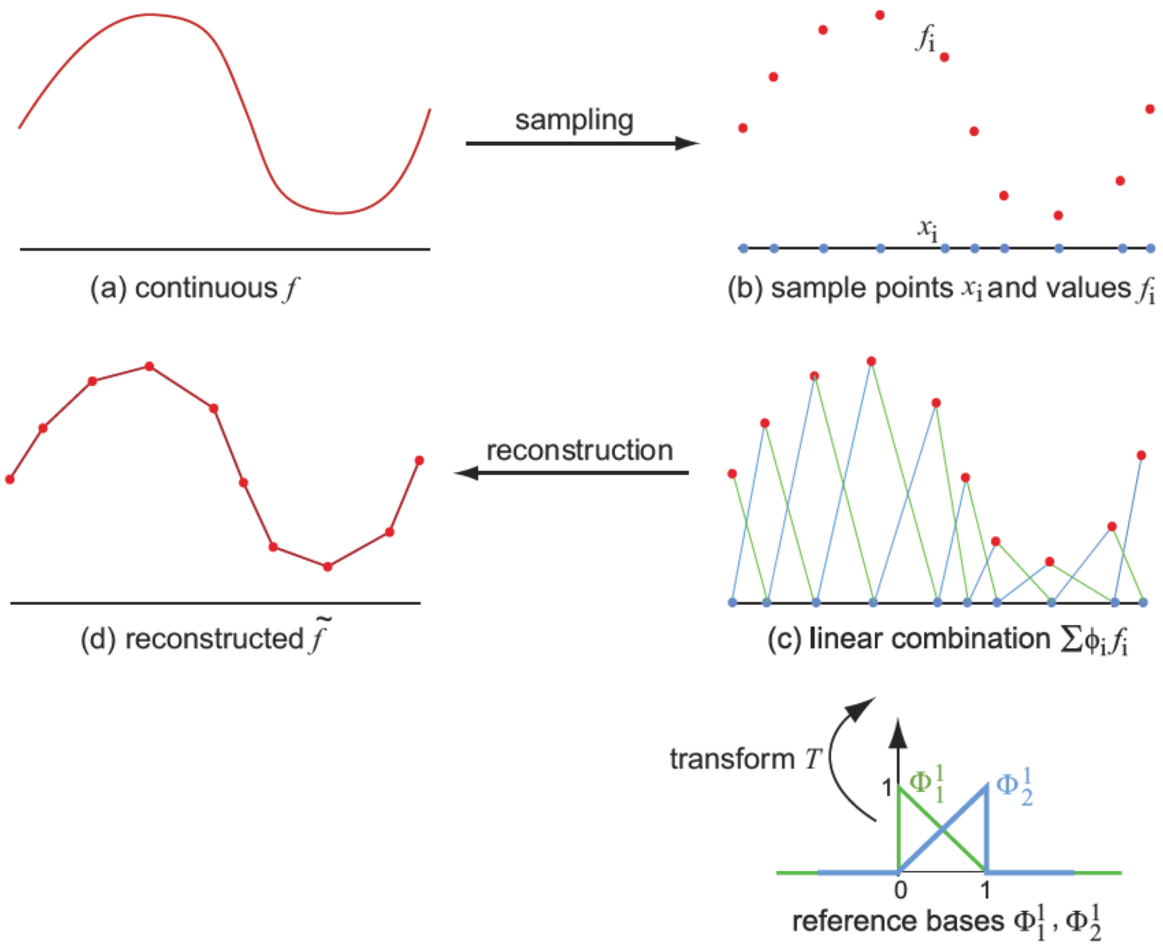
Basis/interpolation functions

- One way to define reconstruction function: $\tilde{f} = \sum_{i=1}^{N} f_i \phi_i$
- Subsequently, we get $\tilde{f}(p_j) = \sum_{i=1}^{N} f_i \phi_i(p_j) = f(p_j) = f_j$ for $\forall j$

- Orthogonality basis function $$\phi_i(p_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

- Normality of basis function $$\sum_{i=1}^{N} \phi_i(x) = 1, \forall x \in \mathrm{D}$$

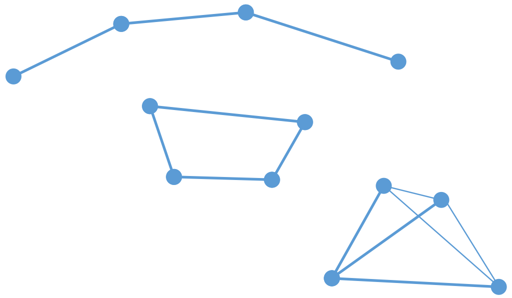
Sampled Data

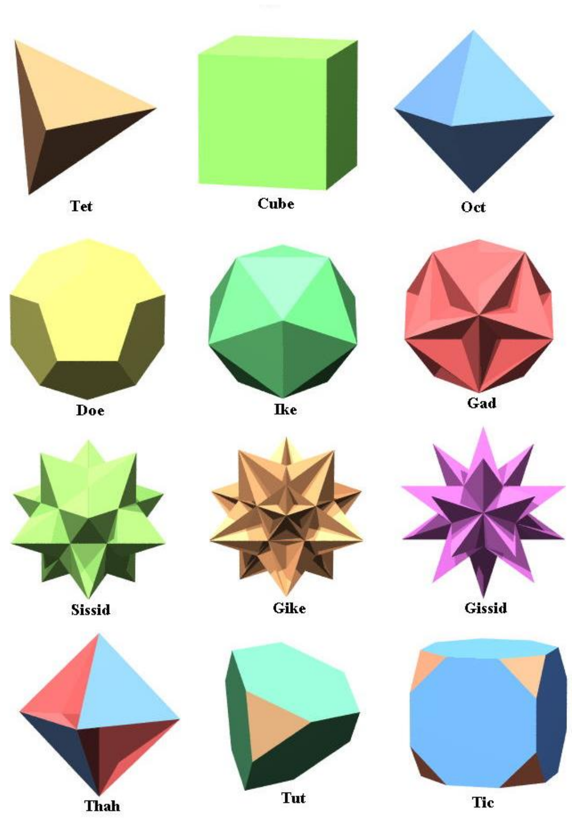


Source: Data Visualization: Principles and Practice

Domain Subdivision

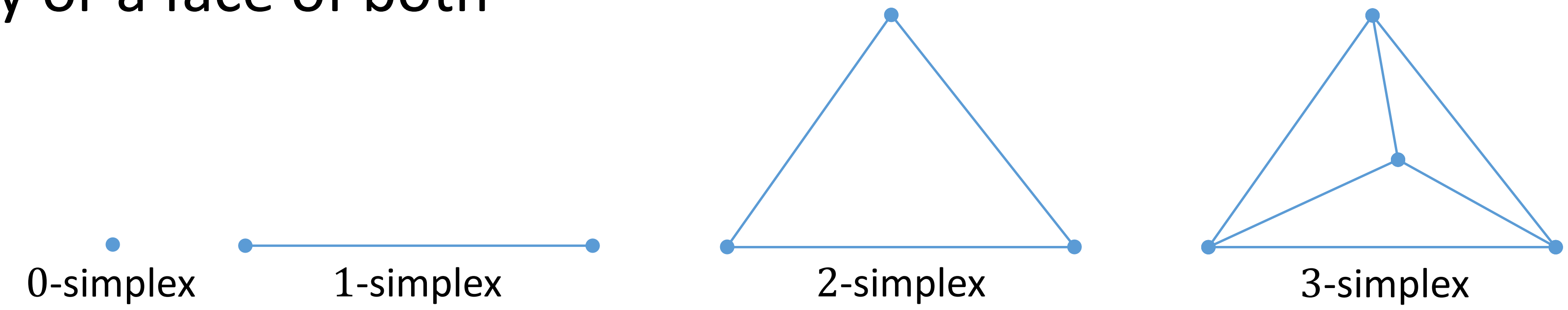
- A grid (aka mesh) is a subdivision of a domain D into a collection of cells (aka elements)
- Most commonly used cells:
  - Polylines in $\mathbb{R}$
  - Polygons in $\mathbb{R}^2$
  - Polyhedra in $\mathbb{R}^3$
- Union of cells cover entire domain D and cells are non-overlapping, and vertices are sample points





Domain Subdivision

- The most common domain discretization used in SciVis are simplicial complexes
- A d-simplex is the convex hull of $d + 1$ affinely independent points in $\mathbb{R}^d$ is d-dimensional
- A simplicial complex is a finite collection of simplices that contains all faces of any simplex and where the intersection of two simplicesis empty or a face of both



Constant Basis Function

constant, zero-order continuity global basis function

$$\phi_i^0(x) = \begin{cases} 1, & x \in c_i \\ 0, & x \notin c_i \end{cases}$$

- Simplest set of basis function:
- Sample points are inside the grid cells
- Nearest-neighbor interpolation
- Virtually no computation cost
- Work with any cell shape and in any dimension
- Provide a poor, staircase-like approximation
- We can provide a better (i.e. more continuous) reconstruction of the original function

Linear Basis Functions

- Linear basis function – next simplest basis functions
- Need to make some assumption about the cell types used in the grid
- Assume quadrilateral cells having 4 vertices
- Reference quad cell in $\mathbb{R}^2$: $v_1 = (0,0), v_2 = (1,0), v_3 = (1,1), v_4 = (0,1)$

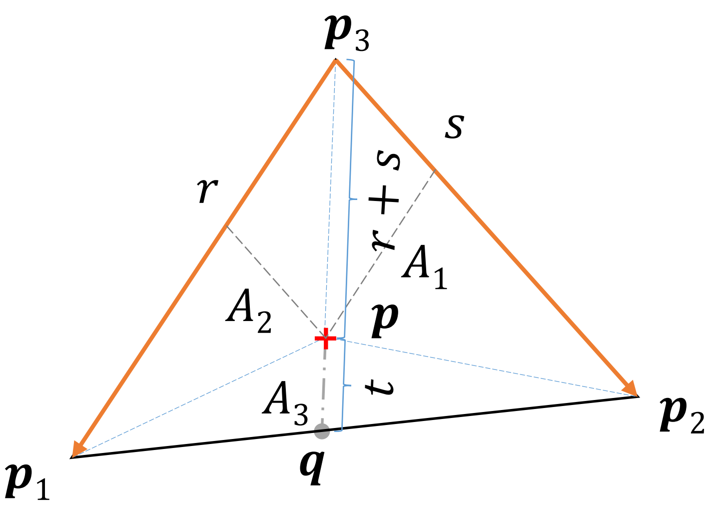
Set of local basis functions

$$
\begin{aligned}
\Phi_1^1(r, s) &= (1-r)(1-s),\\
\Phi_2^1(r, s) &= r(1-s),\\
\Phi_3^1(r, s) &= rs,\\
\Phi_4^1(r, s) &= (1-r)s;
\end{aligned}
$$

reference coordinates

Barycentric Coordinates

- A simplex is a convex hull of $k+1$ points in a k-dimensional space
- Barycentric coordinates provide a simple way to interpolate over simplices
- In case of (planar) triangles, $k=2$



$p1 := \begin{bmatrix} 27.2 \\ 11.8 \end{bmatrix}$ $\quad p2 := \begin{bmatrix} 68.4 \\ 16.3 \end{bmatrix}$ $\quad p3 := \begin{bmatrix} 46.6 \\ 40.9 \end{bmatrix}$

$A1 := 232.2 \quad A2 := 186.24 \quad A3 := 137.37$

$A := A1 + A2 + A3 = 555.81$

$r := \frac{A1}{A} = 0.418 \quad s := \frac{A2}{A} = 0.335 \quad t := \frac{A3}{A} = 0.247$

$r + s + t = 1 \quad 1 - r - s = 0.247$

$p := r \cdot p1 + s \cdot p2 + t \cdot p3 = \begin{bmatrix} 45.8 \\ 20.5 \end{bmatrix}$

$\boxed{p := r \cdot p1 + s \cdot p2 + (1 - r - s) \cdot p3} = \begin{bmatrix} 45.8 \\ 20.5 \end{bmatrix}$

or in the rearranged form

$p := r \cdot p1 + s \cdot p2 + 1 \cdot p3 - r \cdot p3 - s \cdot p3 = \begin{bmatrix} 45.8 \\ 20.5 \end{bmatrix}$

$\boxed{p := r \cdot (p1 - p3) + s \cdot (p2 - p3) + 1 \cdot p3} = \begin{bmatrix} 45.8 \\ 20.5 \end{bmatrix}$

From this equation is clear what the barycentric coordinates r and s actually mean

Barycentric Coordinates for Triangles

- Barycentric (area) coordinates $[r, s, t]$ describe location of a point $\boldsymbol{p}$ in a triangle in relation to vertices $\boldsymbol{v}_i$

$$\boldsymbol{p} \; = \; [r, s, t] \; = r\,\boldsymbol{v}_1 \; + s\,\boldsymbol{v}_2 \; + t\,\boldsymbol{v}_3\,,$$

  where $r, s, t \;\geq\; 0$ and $r + s + t = 1$

- Coordinates corresponds to the signed area of the opposite subtriangle divided by area of the triangle
- Note that the point $\boldsymbol{p}$ is uniquely defined by any two of the three barycentric coordinates, e.g. r and s

$$\boldsymbol{p} \; = \; [r, s] \; = \; r\,\boldsymbol{v}_1 \; + s\,\boldsymbol{v}_2 \; + (1 - r - s)\,\boldsymbol{v}_3 \text{ or } T(r, s) = \sum_{i=1}^{3} \boldsymbol{v}_i \Phi_i^1(r, s)$$

- In the same way, we can interpolate any quantity (or function) inside the triangle

Forward Transformations

- Given any cell type having n vertices $\boldsymbol{p}_i$ in $\mathbb{R}^3$, we define transformation T that maps from a point $[r, s]$ in reference cell coordinate system to a point $[x, y, z]$ in the actual cell as follows

$$\boldsymbol{p} = [x, y, z] = T(r, s) = \sum_{i=1}^{n} \boldsymbol{p}_i \Phi_i^1(r, s)$$

- T maps the reference cell to the world cell
- T^{-1} maps points $[x, y, z]$ in the world cell to points $[r, s]$ in the reference cell

Backward Transformation

- Having T^{-1}, we can rewrite the reconstruction function $\tilde{f} = \sum_{i=1}^{n} f_i \Phi_i$ for quad cell as

$$\tilde{f}(x, y, z) = \sum_{i=1}^{4} f_i \Phi_i^1(T^{-1}(x, y, z))$$

- To compute T^{-1}, we have to invert the expression $T(r, s) = \sum_{i=1}^{n} p_i \Phi_i^1(r, s)$
- Given a rectangular cell, this yields

$$T_{\mathrm{rect}}^{-1}(x, y, z) = (r, s) = \left(\frac{(p - p_1) \cdot (p_2 - p_1)}{\|p_2 - p_1\|^2}, \frac{(p - p_1) \cdot (p_4 - p_1)}{\|p_4 - p_1\|^2} \right)$$

- Now, we have a simple way how to reconstruct a piecewise C^1 function from samples on any rectangular grid. Arbitrary quad cells require some more elaborated numerical solution for obtaining r, s

Backward Transformation for Triangles

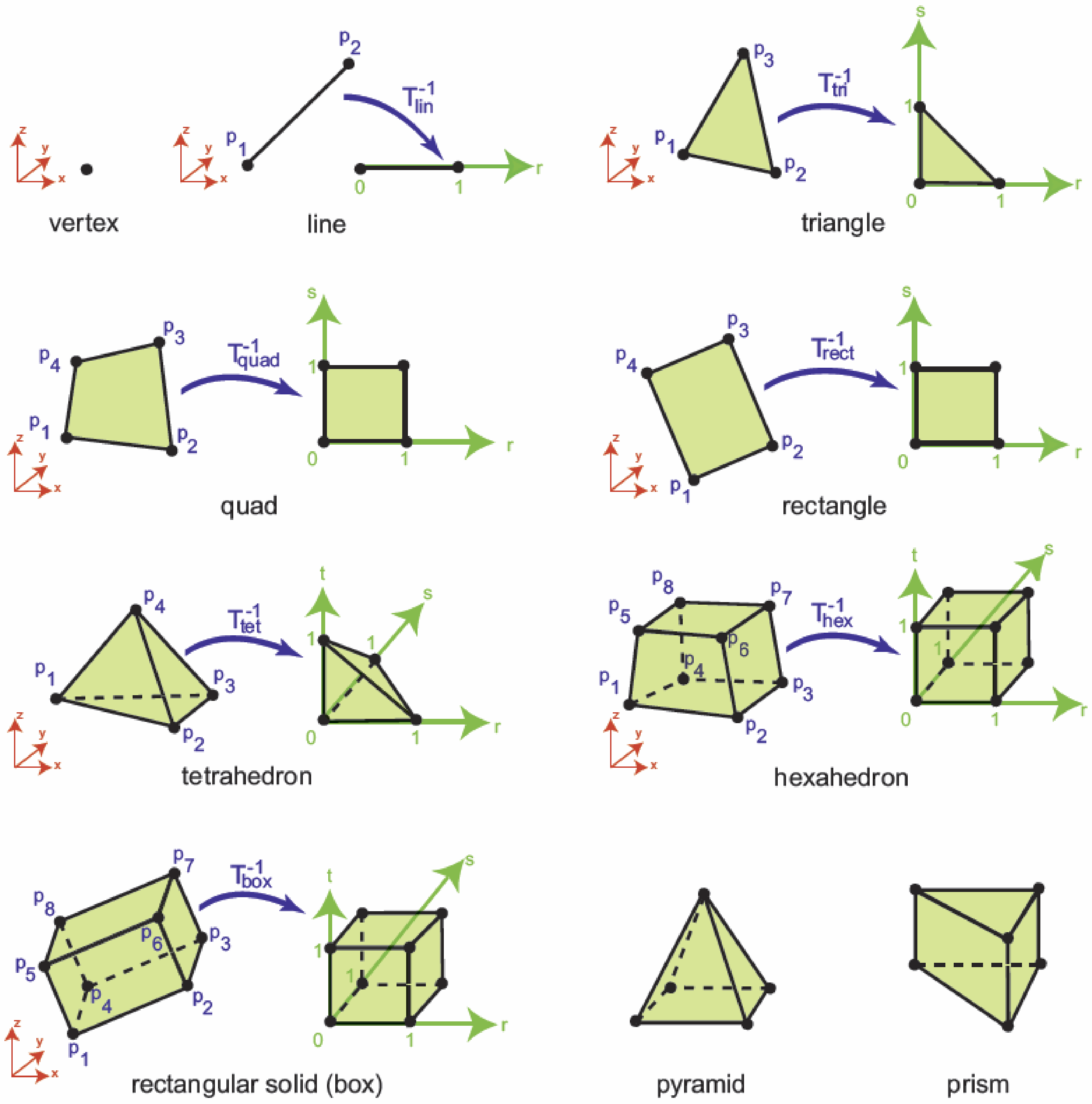
- Same situation but with triangle cell (simplest cell in 2D)
- Three linear basis functions

$$\Phi_1^1(r,s) = r$$
$$\Phi_2^1(r,s) = s$$
$$\Phi_3^1(r,s) = 1 - r - s$$

- The transformation T^{-1} for triangular cells

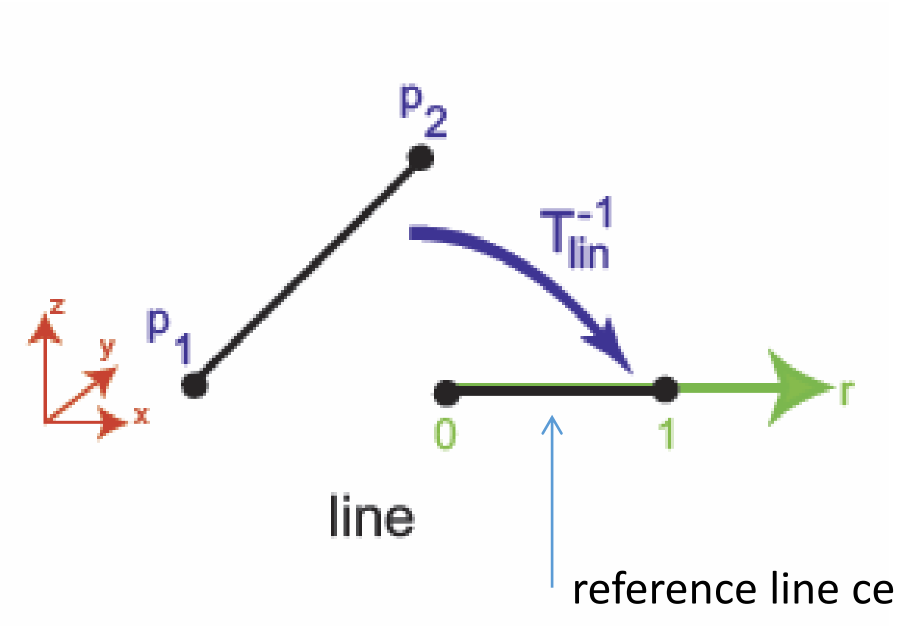
$$T_{\mathrm{tri}}^{-1}(x,y,z) = (r,s) = \left(\frac{\|(p-p_1)\times(p_3-p_1)\|}{\|(p_2-p_1)\times(p_3-p_1)\|}, \frac{\|(p-p_1)\times(p_2-p_1)\|}{\|(p_3-p_1)\times(p_2-p_1)\|} \right)$$

Cells



Source: Data Visualization: Principles and Practice

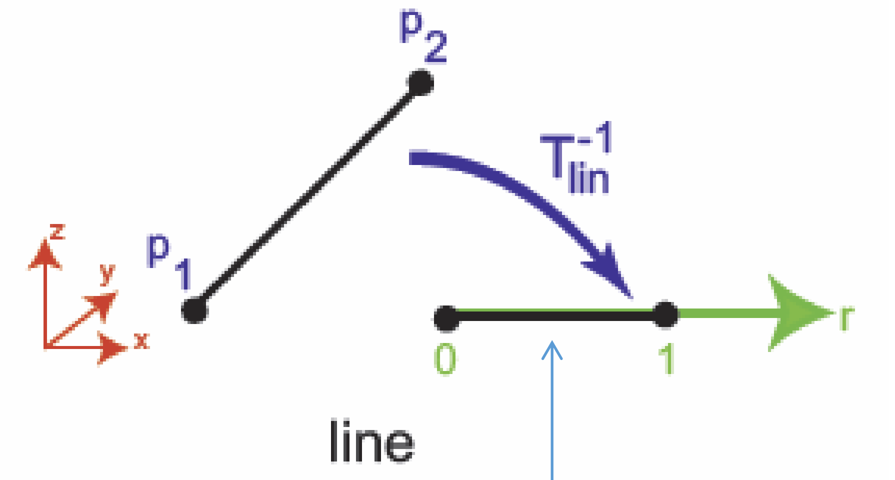
Line Cell



reference line cell

Source: Data Visualization: Principles and Practice

Line Cell



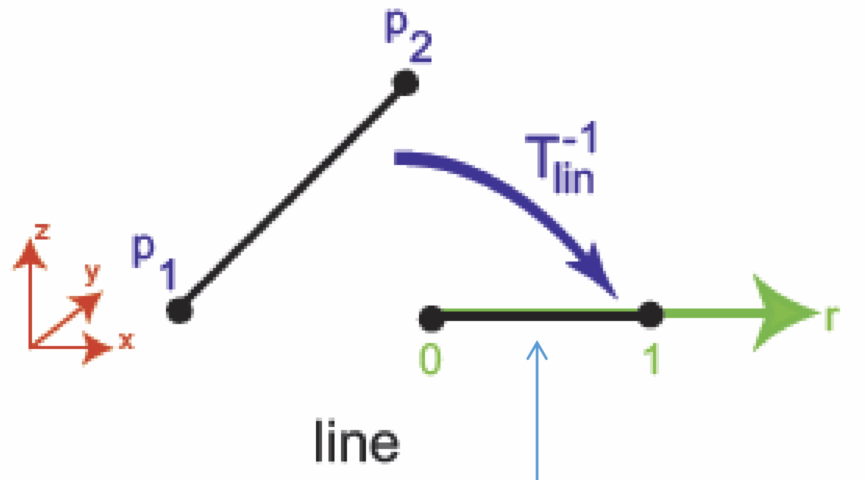
reference line cell

$$\Phi_1^1(r)$$
$$\Phi_2^1(r)$$

$$T_{\mathrm{lin}}^{-1}(x, y, z)$$

Source: Data Visualization: Principles and Practice

Line Cell



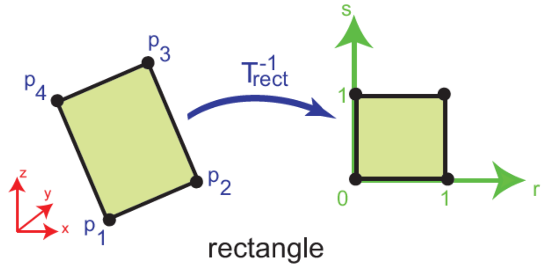
$$\Phi_1^1(r) = 1 - r$$
$$\Phi_2^1(r) = r.$$

two linear basis functions

$$T_{\text{lin}}^{-1}(x, y, z) = \frac{\|p - p_1\|}{\|p_2 - p_1\|}$$

Source: Data Visualization: Principles and Practice

Rectangular Cell



$$\Phi_1^1(r,s) = (1-r)(1-s),$$
$$\Phi_2^1(r,s) = r(1-s),$$
$$\Phi_3^1(r,s) = rs,$$
$$\Phi_4^1(r,s) = (1-r)s.$$

$$T_{\text{rect}}^{-1}(x,y,z) = (r,s) = \left(\frac{(p-p_1)\cdot(p_2-p_1)}{\|p_2-p_1\|^2}, \frac{(p-p_1)\cdot(p_4-p_1)}{\|p_4-p_1\|^2} \right)$$

$$\Phi_1^1(r,s) = (1-r)(1-s),$$
$$\Phi_2^1(r,s) = r(1-s),$$
$$\Phi_3^1(r,s) = rs,$$
$$\Phi_4^1(r,s) = (1-r)s.$$

General Quad Cell

- The world coordinates of a point $\boldsymbol{p}$ on the given quadrilateral cell with known parametric coordinates r, s can be computed as follows

$$\boldsymbol{p} = (x,y,z)^T = T_{quad}(r,s) = \sum_{i=0}^{4} \boldsymbol{p}_i \Phi_i^1(r,s)\ ,$$

where T_{quad} is a simple bilinear interpolation on a rectangle

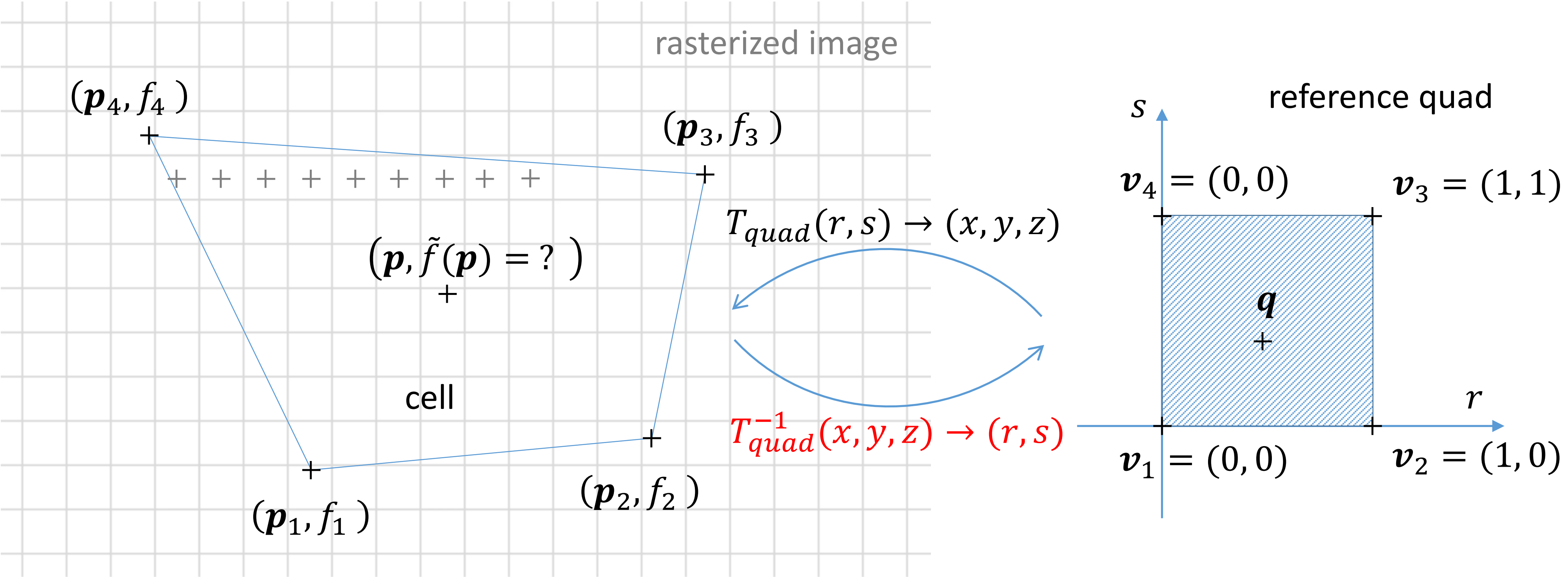
$$T_{quad}(\mathrm{r},\mathrm{s}) = (1-\mathrm{s})[(1-\mathrm{r})\boldsymbol{p}_1 + r\boldsymbol{p}_2] + \mathrm{s}[r\boldsymbol{p}_3 + (1-\mathrm{r})\boldsymbol{p}_4]$$

- Typically, we would like to obtain interpolated quantity f at this point

$$\tilde{f}(x,y,z) = \sum_{i=1}^{4} f_i\, \Phi_i^1(T_{quad}^{-1}(x,y,z))$$

Unlike in previous cases, we cannot guess inverse of T_{quad} directly.

General Quad Cell



We are looking for the interpolated valued of f at the point $\boldsymbol{p}$ and to do so, we need to find the corresponding point $\boldsymbol{q}$ in reference coordinates to be able to evaluate $\tilde{f}(\boldsymbol{p})$.

General Quad Cell

To find the zeroes of a single vector-valued function $\boldsymbol{F}: \mathbb{R}^k \to \mathbb{R}^{k\ or\ m}$ we may use Newton's method: $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{J}_F(\boldsymbol{x}_n)^{-1}\boldsymbol{F}(\boldsymbol{x}_n)$

- One solution is to numerically solve for r, s as functions of $x, y(, z)$ by Newton's method (note that $\boldsymbol{T}_{quad} = \boldsymbol{T}: \mathbb{R}^2 \to \mathbb{R}^3$ here)

$$\begin{pmatrix} r^{t+1} \\ s^{t+1} \end{pmatrix} = \begin{pmatrix} r^t \\ s^t \end{pmatrix} - \boldsymbol{J}_T^{-1}(r^t, s^t)(\boldsymbol{T}(r, s) - \boldsymbol{p})$$

where $\boldsymbol{J}_T^{-1}$ is the generalized inverse of the non-square Jacobian matrix

Also note that matrix J_T evolves over time as well as r and s get updated during iterations

$$\boldsymbol{J}_T(r, s) = \begin{pmatrix} \vdots & \vdots \\ \frac{\partial \boldsymbol{T}}{\partial r}(r, s) & \frac{\partial \boldsymbol{T}}{\partial s}(r, s) \\ \vdots & \vdots \end{pmatrix}_{2\ or\ 3\times 2}$$

Matrix describes direction and speed of position changes of $\boldsymbol{T}$ when r, s are varied.

General Quad Cell

- It is easy to see that

$$\frac{\partial \boldsymbol{T}}{\partial r}(r,s) = (s-1)(\boldsymbol{p}_1 - \boldsymbol{p}_2) + s(\boldsymbol{p}_3 - \boldsymbol{p}_4)$$

and

$$\frac{\partial \boldsymbol{T}}{\partial s}(r,s) = (r-1)(\boldsymbol{p}_1 - \boldsymbol{p}_4) + r(\boldsymbol{p}_3 - \boldsymbol{p}_2)$$

- The pseudo inverse of $\boldsymbol{J}_T$ can be computed by function

```
cv::invert( J, J_inv, cv::DECOMP_SVD ); // C++ (but it is terribly slow)
J_inv = numpy.linalg.pinv( J ) # Python
```

Example on General Quad Cell

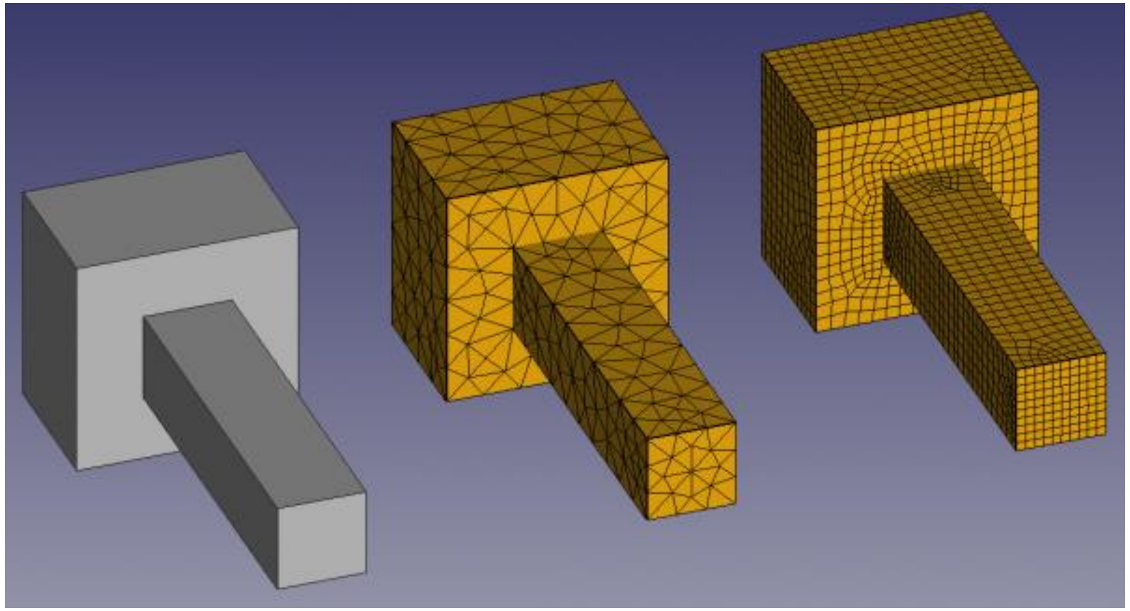
- Test example for the afore described procedure:

For the cell with vertices $\boldsymbol{p}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \boldsymbol{p}_2 = \begin{pmatrix} 3 \\ 0.25 \\ 1 \end{pmatrix}, \boldsymbol{p}_3 = \begin{pmatrix} 4 \\ 3 \\ 1 \end{pmatrix}, \boldsymbol{p}_4 = \begin{pmatrix} 0 \\ 3.5 \\ 1 \end{pmatrix}$ and the query point $\boldsymbol{p} = \begin{pmatrix} 1.8 \\ 2.7 \\ 1 \end{pmatrix}$ we get $r = 0.445$ and $s = 0.818$ just after 3 iterations while the initial estimates of r^0 and s^0 are set to 0.5.

- If you get the same r and s for given vertices, your implementation is probably correct

Exercise

- Typically, finite-element meshes are generated from constructive-solid-geometry (CSG) models during finite element analysis (FEA) used in engineering
- Triangular and quadrilateral subdivisions of simulation domains are the most common



(a) CSG model (b) Triangular mesh (c) Quadrilateral mesh